MATTHEW D. POWERS (Bar No. 104795)
E-Mail: matthew.powers@weil.com
EDWARD R. REINES (Bar No. 135960)
E-Mail: edward.reines@weil.com
JEFFREY G. HOMRIG (Bar No. 215890)
E-Mail: jeffrey.homrig@weil.com
JILL J. HO (Bar No. 236349)
E-Mail: jill.ho@weil.com
WEIL, GOTSHAL & MANGES LLP
Silicon Valley Office
201 Redwood Shores Parkway
Redwood Shores, CA 94065
Telephone: (650) 802-3000
Facsimile: (650) 802-3100

Attorneys for Plaintiff-Counterclaim Defendant
NETAPP, INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

NETWORK APPLIANCE, INC., Plaintiff-Counterclaim Defendant, v. SUN MICROSYSTEMS, INC., Defendant-Counterclaim Plaintiff.	Case No. C-07-06053-EDL **PLAINTIFF NETAPP, INC.'S RESPONSE TO SUN'S OPENING CLAIM CONSTRUCTION BRIEF** Hon. Elizabeth D. Laporte Complaint filed: September 5, 2007 Hearing Date: August 27, 2008

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

CASES

TABLE OF AUTHORITIES
(continued)

Page(s)

STATUTES

TABLE OF AUTHORITIES
(continued)

Page(s)

MISCELLANEOUS

I.

INTRODUCTION

NetApp explains in this brief why the Court should adopt its constructions of the disputed terms from United States Patent Nos. 5,819,292 ("the '292 patent"), 6,892,211 ("the '211 patent"), and 7,200,715 ("the '715 patent"). *See* Declaration of Jill J. Ho (hereinafter "Ho Decl.") Exhs. A-C. Because NetApp's constructions are supported by the intrinsic evidence, and indeed all the evidence, and because Sun repeatedly attempts to import unwarranted limitations from the preferred embodiments, the Court should adopt NetApp's constructions and reject Sun's constructions.

II.

NETAPP'S CONSTRUCTIONS ARE SUPPORTED BY THE EVIDENCE

A. U.S. PATENT NO. 5,819,292

1. "non-volatile storage means"

Term	NetApp's Construction	Sun's Construction
"non-volatile storage means"	A storage device that can retain information in the absence of power.	This limitation is a means-plus-function limitation governed by 35 U.S.C. § 112, ¶ 6. <u>Function</u>: storing blocks of data of a file system so that the data is not lost in the absence of power <u>Structure</u>: one or more disks with a block-based format (i.e., 4KB blocks that have no fragments), where the disk storage blocks are the same as the data blocks of the file system

The dispute between the parties is whether the term "non-volatile storage" – which has a simple, well understood meaning in the art – should be contorted to require a variety of highly specific file system characteristics such as blocks that are fixed "4 KB blocks that have no fragments." Sun contends that, under 35 U.S.C. § 112, ¶ 6, the well-understood, structural implications of this term should be disregarded because the term is used in connection with the word "means." But, as explained below, the claim term is not even a means-plus-function

limitation. And, even if it were, unambiguous guidance in the specification makes clear that the term should not be so narrowly understood.

a. "Non-Volatile Storage Means" Is Not A Means-Plus-Function Limitation Because It Recites No Function

Latching on to the fact that the term "non-volatile storage" is used in connection with the word "means," Sun starts by erroneously assuming that the term is *presumed* to be a means-plus-function limitation. But there is no such presumption here. The present circumstances simply are not the traditional circumstances that the rule creating the presumption was intended to encompass. Indeed, the claims at issue here are process claims, not apparatus claims. This is important because the rule creating the presumption on when § 112, ¶ 6 is invoked has at its roots the long-standing, well known principle that in the context of apparatus claims the "means . . . for" language is used to invoke § 112, ¶ 6, while in the context of method claims the "step . . . for" language is used. *See, e.g., O.I. Corp. v. Tekmar Co.*, 115 F.3d 1576, 1582-1583 (Fed. Cir. 1997) ("[T]he word 'means' clearly refers to the generic description of an apparatus element, and the implementation of such a concept is obviously by structure or material. We interpret the term 'steps' to refer to the generic description of elements of a process, and the term 'acts' to refer to the implementation of such steps.") (citations omitted). The claims at issue here simply include the word "means" buried in a step of a *process* claim for a *method* of maintaining a file system. And, instead of being connected to a particular function through the preposition "for," the word "means" is joined to the inherently structural term "non-volatile storage." This use of the word "means" simply does not present the type of facts that brought about the rule creating a presumption that a term is a means-plus-function limitation.

Even if the presumption did apply, though, "non-volatile storage" means still would not be considered a means-plus-function limitation. "[T]he mere use of the word 'means' after a limitation, without more, does not suffice to make that limitation a means-plus-function limitation." *Allen Eng'g Corp. v. Bartell Indus.*, 299 F.3d 1336, 1347 (Fed. Cir. 2002); *see also Cole v. Kimberly-Clark Corp.*, 102 F.3d at 531 ("Merely because a named element of a patent claim is followed by the word "means," however, does not automatically make that element a

"means-plus-function" element . . . ."). Indeed, "a claim element that uses the word 'means' but recites no function corresponding to the means does not invoke § 112, ¶ 6." *Allen Eng'g*, 299 F.3d at 1347 (citations omitted). Contrary to Sun's urging, there is no function corresponding to the "non-volatile storage means."

In its brief, Sun urges that "non-volatile storage means" corresponds to three distinct functions recited in two different independent claims of the '292 patent. For example, Sun points to the first limitation of Claim 4, which recites "storing a first file system information structure . . . in said non-volatile storage means." *See* Ho Decl., Exh. A ('292 patent) at 25:16. However, the "non-volatile storage means" is not explicitly linked to this function through the preposition "for," and, in fact, does not perform the "storing" function. Rather, the thing carrying out the "storing" is the component implementing the claimed methods (*e.g.*, software executing on a file server), which merely happens to store things "*in*" the "non-volatile storage." *See* Ganger Decl. ¶ 11. Indeed, by using the word "in," the claim confirms that the "non-volatile storage means" is a passive component into which something is stored, rather than an active component performing the described step of "storing." That the "non-volatile storage means" is not the thing doing the "storing" is confirmed elsewhere in the claim. Specifically, the very next limitation in the claim requires "writing blocks of data of said file system . . . to said non-volatile storage means," again showing that the "non-volatile storage" means is being acted on. *See* Ho Decl., Exh. A ('292 patent) at 25:21-24. Further, the claims contain nothing to combat the natural interpretation that the thing doing the "storing" is different from the thing doing the "writing." But the "writing" is a task that a "non-volatile storage means" definitely does not do. *See id.* ¶ 11. This is because the "non-volatile storage means" is the thing that is being *written to*. Thus, both the "writing" and "storing" are done by the components implementing the claimed method, which is *not* the "non-volatile storage means" that is *acted upon* in carrying out the method. *See id.* ¶ 11. These arguments apply equally to the other functions that Sun erroneously assigns to the function of the "non-volatile storage means."

b. "Non-Volatile Storage Means" Is Not A Means-Plus-Function Limitation Because It Recites Sufficient Structure

"[E]ven if the claim element specifies a function, if it also recites sufficient structure or material for performing that function, § 112, ¶ 6 does not apply." *Allen Eng'g*, 299 F.3d at 1347. In determining whether a claim term recites sufficient structure to take it out of § 112, ¶ 6, the bar is low: "[a] claim term recites sufficient structure if the term, as the name for structure, has a reasonably well understood meaning in the art." *Id.* (internal citations and quotations omitted). That bar is well exceeded here.

There can be no doubt that the term "non-volatile storage" has a reasonably well understood meaning in the art. Indeed, in the Joint Claim Construction and Prehearing Statement, Sun cited to no less than seven dictionary definitions proving this to be the case. Every single one of Sun's definitions was consistent in defining "non-volatile storage" or "non-volatile memory" to be a storage device that retains data in the absence of power, which is precisely the construction NetApp proposes, and all that should be required of this non-means-plus-function term. *See* Ho Decl., Exh. D (Supp. Joint Cl. Const. and Prehearing Statement), Exh. A. Furthermore, a number of Sun's dictionary definitions point to specific electronic components, demonstrating that structure is inherent in this term. For instance, the *Microsoft Press Computer Dictionary* explains that "nonvolatile memory" is "[i]ntended to refer to core, ROM, EPROM, bubble memory, or battery backed CMOS RAM" and that "the term is occasionally used in reference to disk subsystems as well." *See* Ganger Decl., Exh. F (*Microsoft Press Computer Dictionary*, 241 (1991)); *see also* Brandt Decl., Exh. 3 (*Webster's New World Dictionary of Computer Terms*, 376 (8th ed. 2000)) (in describing "nonvolatile memory," explaining that "[r]ead only memory (ROM) is nonvolatile, as are all secondary storage units such as disk drives").[1] .

---

[1] In his declaration, NetApp's expert, Professor Ganger, identifies yet additional dictionary definitions and explains how they affirm that "non-volatile storage" is a storage device that retains data in the absence of power. Professor Ganger further points out how many of these definitions point to specific storage components, thus confirming the structural nature of the term "nonvolatile storage" *See* Ganger Decl. ¶ 12. For instance, Professor Ganger notes that the *The Illustrated Dictionary of Computer Words* defines "non-volatile storage" to be, in part, "[s]torage medium that retains its data in the absence of power, such as *magnetic bubble memory* and *magnetic core storage*." Ganger Decl., Exh. L.

As the Federal Circuit explained in *Greenberg v. Ethicon Endo-Surgery*, this is compelling evidence rebutting Sun's position that "non-volatile storage" must be treated under 35 U.S.C. § 112, ¶ 6. In *Greenberg v. Ethicon Endo-Surgery*, 91 F.3d 1580, 1583 (Fed. Cir. 1996), the Federal Circuit considered whether the term "detent mechanism" was a means-plus-function limitation. In holding that it was not, the Federal Circuit explained as follows:

> Dictionary definitions make clear that the noun "detent" denotes a type of device with a generally understood meaning in the mechanical arts, even though the definitions are expressed in functional terms. It is true that the term "detent" does not call to mind a single well-defined structure, but the same could be said of other commonplace structural terms such as "clamp" or "container." What is important is not simply that a "detent" or "detent mechanism" is defined in terms of what it does, but that the term, as the name for structure, has a reasonably well understood meaning in the art.

Greenberg, 91 F.3d 1583 (internal citations and quotations omitted). Here, as in *Greenberg*, the term "non-volatile storage" does not necessarily "call to mind a single well-defined structure," but the term nonetheless has a "reasonably well understood meaning in the art" as the name for structure. Indeed, every single dictionary definition the parties are aware of at least defines "non-volatile storage" or "non-volatile memory" in "functional terms" to be a storage device that retains information in the absence of power. However, unlike as in *Greenberg*, the definitions of "non-volatile storage" here are not limited to mere "functional terms," but often include specific structure as well, confirming that "non-volatile storage" connotes specific structure such that § 112, ¶ 6 does not apply. Under similar circumstances, Courts have repeatedly concluded that a similarly innate structural term does not implicate § 112, ¶ 6. *See, e.g.*, *Cole*, 102 F.3d at 531 (using a dictionary definition as an aid to conclude that "perforation means" did not invoke § 112, ¶ 6); *Envirco Corp. v. Clestra Cleanroom, Inc.*, 209 F.3d 1360, 1365 (Fed. Cir. 2000) (using a dictionary definition as an aid to conclude that "baffle means" did not invoke § 112, ¶ 6); *Keithley v. Homestore.com, Inc.*, 2007 WL 2701337, at *20 (N.D. Cal. Sep. 12, 2007) (using dictionary definitions as an aid to conclude that "file serve means" did not invoke § 112, ¶ 6).

Notably, at least one district court has even considered whether the term "storage means" implicates § 112, ¶ 6. In *Lottotron, Inc. v. Scientific Games Corp.*, 2003 WL 22075683 (S.D.N.Y. 2003), the claim term at issue was "storage means connected to said message means for receiving and storing said wagering information." In holding that this was not a means-plus-

function limitation, the Court pointed to a technical dictionary definition explaining that storage was "[a] device consisting of electronic, electrostatic, electrical, hardware or other elements into which data may be entered, and from which data may be obtained as desired." *Lottotron*, 2003 WL 22075683 at *7. The Court explained that the use of the term "means" did "little to diminish the structural character of this element." *Id.* The term at issue here is an even less likely candidate for § 112, ¶ 6 treatment. Unlike as in *Lottotron*, the disputed claim term here is not connected to a function through the preposition "for," as is traditionally done to invoke § 112, ¶ 6, and is an even more precise recitation of the structure than the term "storage means" at issue there. Thus, if the term "storage means" in *Lottotron* was not a means-plus-function limitation, the term "nonvolatile storage means" at issue here is most definitely not.

c. Sun's Brief Shows That "Non-Volatile Storage" Corresponds To Well Understood Structure

The above evidence and authority itself confirms that "non-volatile storage" is a term that is adequately understood in the art such that § 112, ¶ 6 is not implicated. But if this was not enough, Sun's Opening Claim Construction Brief provides yet additional evidence confirming this. Indeed, in explaining the technology behind the '211 patent, Sun, citing the declaration of its expert, exhibits comfort with the similar term "permanent storage," noting that "data is written to *permanent storage, such as a disk*." Sun's Opening Cl. Constr. Br. at 19. The declaration of Sun's expert, Dr. Brandt, nonchalantly repeats this language almost word-for-word. *See* Brandt. Decl., ¶ 101. This is proof-positive that someone of skill in the art – for instance, Dr. Brandt – understands full well that when the '292 patent (which shares the same specification as the '211 patent) refers to "non-volatile storage" it must be referring to a particular structural component, "such as a disk."[2] *See* Ganger Decl. ¶ 14. Likewise, Sun's proposed construction for the claim

[2] Furthermore, as Professor Ganger explains in his declaration, Marshall McKusick, another expert that Sun retained in this matter (and who Sun initially selected to testify on the meaning of "non-volatile storage") also seems to be comfortable identifying a structure for the term. *See* Ganger Decl., ¶ 15; Ho Decl., Exhs D, E and F. Indeed, Professor Ganger explains that in a very recent publication describing the development of the BSD file system, Mr. McKusick equated "nonvolatile storage" with "disk," stating that "information in nonvolatile storage (i.e., disk) must always be consistent . . . ." *See* Ganger Decl., Exh. M (Marshall Kirk McKusick, *A Brief History of the BSD Fast File System*, ;login: The Usenix Magazine, June 2007) at 12. Likewise, Professor Ganger explains that in a publication he co-authored with Mr. McKusick, "nonvolatile storage" was described in an identical way. *See* Ganger Decl., Exh. N (Ganger et al., *Soft Updates: A*

term "file system information structure," which Dr. Brandt agrees with, calls for the "file system information structure" to be placed "in a fixed location *on disk*." *See* Brandt Decl., ¶ 77. But, according to Sun, one function for the "nonvolatile storage means" is storing "file system information structures." *See* Sun's Opening Cl. Constr. Br. at 8. Thus, both Sun and Dr. Brandt appear to have no trouble equating the "nonvolatile storage" to a computer disk, which is in fact one way that someone of ordinary skill in the art would understand the term.

Indeed, as Professor Ganger opines in his declaration, Sun's own patent portfolio reveals many patents – too many to list exhaustively – in which Sun apparently had no trouble understanding the meaning of "non-volatile storage" and equating it with a structure, generally a computer disk:

- "The IN-Mod must be updated (for a given inode) before it is commited to *non-volatile storage (i.e. disk or NVRAM)* . . . ." Ganger Decl., Exh. O (U.S. Patent No. 7,089,293) at 47:13-14.
- "*[N]on-volatile storage* . . . may include a floppy disk drive, a RAM card, a hard drive, CD-ROM drive, or other magnetic, re-writable optical, or other mass storage devices" Ganger Decl., Exh. P (U.S. Patent No. 5,954,826) at 6:66-7:2.
- "In the server, there is a large capacity *non-volatile storage device, such as a hard disk drive* . . . ." Ganger Decl., Exh. Q (U.S. Patent No. 5,721,824) at 1:22-23.
- "[A] data storage system includes a computer coupled to a *non-volatile storage, such as a disk drive* . . . ." Ganger Decl., Exh. R (U.S. Patent No. 6,629,198) at 2:2-4.
- "[T]he processor's internal context is saved in RAM or non-volatile memory (e.g., disk storage) . . . ." Ganger Decl., Exh. S (U.S. Patent No. 5,878,264) at 10:28-29.
- "Storage 164, such as a computer disk drive or other nonvolatile storage may provide storage of data or program instructions." Ganger Decl., Exh. T (U.S. Patent No. 5,929,792) at 4:35.

See Ganger Decl. ¶ 16.

Sun's brief, again citing Dr. Brandt's declaration, even lists a number of structures that could qualify as "non-volatile storage." *See id.* at 7. Sun urges that because it was able to construct a broad list, § 112, ¶ 6 must apply.[3] But this merely proves that the term has a broad

---

Solution to the Metadata Update Problem in File Systems, ACM Transactions on Computer Systems 127 (2000)) at 128 ("[T]he information in nonvolatile storage (*i.e.*, disk) must always be consistent . . . ."). *See id.* ¶ 15.

[3] In an attempt to show that the term "non-volatile storage means" is too broad to be meaningfully understood by someone of skill in the art, Sun, and its expert, include some unusual items in its

meaning, not that it is a means-plus-function limitation. *See Linear Tech. Corp. v. Impala Linear Corp.*, 379 F.3d 1311, 1322 (Fed. Cir. 2004) ("That the disputed term is not limited to a single structure does not disqualify it as a corresponding structure, as long as the class of structures is identifiable by a person of ordinary skill in the art."); *Optimal Rec. Solutions, LLP v. Leading Edge Techs., Inc.*, 6 Fed. App'x 873, 878 (Fed. Cir. 2001) ("The fact that [the terms 'global positioning receiver,' 'display,' and 'memory'] may be broad does not detract from their well understood meanings as the names for structure."); *Greenberg*, 91 F.3d at 1583 (an inherently structural term need not "call to mind a single well-defined structure").

d. Sun's Construction Is Inappropriately Narrow

As explained above, it is a stretch to even understand the term "non-volatile storage" to be a means-plus-function limitation in the first place. Nevertheless, even if it were a means-plus-function limitation, Sun's construction would still be incorrect. Sun's proposed construction – including highly particularized requirements like a "4 KB block size" with "no fragments" – is impermissibly restrictive. As the Federal Circuit explained, "[u]nder § 112, ¶ 6, a court may not import functional limitations that are not recited in the claim, or structural limitations from the written description that are unnecessary to perform the claimed function." *Wenger Mfg., Inc. v. Coating Mach. Sys., Inc.*, 239 F.3d 1225, 1233 (Fed. Cir. 2001); *see also Asyst Techs., Inc. v. Empak, Inc.*, 268 F.3d 1364, 1369-1370 (Fed. Cir. 2001) ("Section 112 paragraph 6 does not permit incorporation of structure from the written description beyond that necessary to perform the claimed function.") (citations omitted); *Chiuminatta Concrete Concepts, Inc. v. Cardinal Indus.*, 145 F.3d 1303, 1308 (Fed. Cir. 1998) (explaining that "additional structural aspects" "unrelated to the recited function" "are not what the statute contemplates as structure *corresponding* to the recited function"). As set forth below, Sun not only attempts to import unnecessary limitations, but attempts to import unnecessary limitations that are unrelated to the concept of a "non-volatile storage" device.

---

list, such as "paper" and "film." *See* Sun's Opening Cl. Constr. Br. at 7; Brandt Decl., ¶ 65. However, someone of skill in the art would understand that in the context of the patent, which is computer file systems, the term "non-volatile storage" simply does not refer to things like "paper" and "film" which are not used in the art for maintaining computer file systems. *See* Ganger Decl., ¶ 17.

(1) **Sun Attempts To Import Limitations That Are Unnecessary To The Invention And Claimed Function**

As explained above, it is a stretch to even understand the term "non-volatile storage" to be a means-plus-function limitation in the first place. Nevertheless, even if it were a means-plus-function limitation, Sun's construction would still be incorrect.

First, if "non-volatile storage means" is purely functional – as Sun erroneously contends – then the function performed is merely "non-volatile storage," because the term "non-volatile storage means" is equivalent to "means for non-volatile storage." *See Signtech USA, Ltd. v. Vutek, Inc.*, 174 F.3d 1352, 1356 (Fed. Cir. 1999) ("[T]he claim element 'ink delivery means' uses the term 'means' in association with a function, namely 'ink delivery.' Although the phrase 'means for' is not used, the phrase 'ink delivery means' is equivalent to the phrase 'means for ink delivery,' because 'ink delivery' is purely functional language."). The meaning of "non-volatile storage" in the art is clear: It is simply a storage device that can retain information in the absence of power. Sun cited seven dictionary definitions confirming this. *See* Ho Decl., Exhs. D and E (Joint and Supp. Joint Cl. Constr. and Prehearing Statements), Exh. A; Ganger Decl. ¶ 12. Thus, to the extent this term requires a corresponding function, that function is nothing more than storing information so that the information is not lost in the absence of power. Sun's added requirement that the function include "blocks of data in a file system" is entirely unwarranted, and, in any event, is addressed expressly in other claim limitations.

To the extent this function requires a corresponding structure, the claims should not be limited to a particularized "nature of disks" that requires such technicalities as (1) 4KB blocks, (2) blocks with no fragments, and/or (3) disk storage blocks with the same size as the file system data blocks. *See id.* ¶ 18-19. Indeed, even under Sun's overly-restrictive definition of the claimed function ("storing blocks of data of a file system so that the data is not lost in the absence of power") these details would not be required to carry out that function. *See id.* ¶ 22. To the contrary, all that is required is any of a set of well-known devices upon which file systems may be maintained, including disks, disk arrays, flash memory drives, and the like. *See id.* ¶ 17 and 23. Sun's attempt to import into the claims numerous additional limitations, unrelated and

unnecessary to the claimed function, is thus improper. *See Wenger*, 239 F.3d at 1233 ("A] court may not import functional limitations that are not recited in the claim, or structural limitations from the written description that are unnecessary to perform the claimed function."); *Golight, Inc. v. Wal-Mart Stores, Inc.*, 355 F.3d 1327, 1334 (Fed. Cir. 2004) ("Moreover, to the extent the assembly contains particular structures for permitting rotation through 360 degrees, such as the follower pin 64 and slot 65, these structures are superfluous to our claim construction analysis because they are not required for performing the claimed function.").

(2) **Sun Attempts To Import Inapposite Limitations Into Its Construction Of "Non-Volatile Storage Means"**

Not only are the additional requirements that Sun seeks to import into the claims – (1) 4KB blocks, (2) blocks with no fragments, and (3) disk storage blocks with the same size as the file system data blocks – unnecessary for the performance of the claimed function, but these limitations do not even pertain to a "non-volatile storage" device, making Sun's proposed construction especially improper. Put simply, Sun is attempting to take limitations from the specification pertaining to how the *file system* organizes the raw capacity of a "non-volatile storage" device (*e.g.*, a disk) and apply them to the underlying device. *See* Ganger Decl., ¶¶ 19-20. This is evident because Sun draws its proposed corresponding structure from a section of the specification describing the "*File-system* Layout" *See* Ho Decl., Exh. A ('292 patent) at 5:47-60; Sun's Opening Cl. Constr. Br. at 9; Ganger Decl., ¶¶ 19-20.

Sun's approach makes little sense. For one thing, Sun has selected only bits and pieces of the *file system* layout description to import onto its construction of the "non-volatile storage" blocks. The complete sentence from the "File-System Layout" section – which Sun omits from its brief and from which Sun admittedly draws its proposed corresponding structure – is as follows:

> "The present invention uses a Write Anywhere File-system Layout (WAFL). This disk format system is block based (i.e., 4 KB blocks that have no fragments), *uses inodes to describe its files, and includes directories that are simply specially formatted files*.

Ho Decl., Exh. A ('292 patent) at 5:49-52. Sun excludes completely from its proposed corresponding structure the italicized text, and offers no explanation for its omission. But the

reason for Sun's exclusion of the italicized text is no big mystery: Sun could not cite to the italicized text – dealing with "inodes," "files," and "directories" – because Sun could not stretch that text to correspond to the blocks of a "non-volatile storage" device. *See* Ganger Decl. ¶ 20. Indeed, as the section header and text show, all of the characteristics in this passage – including the phrase "4KB blocks that have no fragments" – refer to characteristics of the file system, and not its underlying "non-volatile storage means." *See* Ganger Decl. ¶¶ 19-20. The concept of block "fragments" is a well-understood file system concept, not a characteristic of a disk or other "non-volatile storage means." A file system that supports fragments uses them to allow the contents of multiple small files to be packed into a single block – the block is divided into fragments that can each be allocated separately. *See id.* ¶ 20. Sun's attempt to apply such concepts to the disk blocks of a "non-volatile storage" device should thus be rejected.

Put simply, Sun's proposed construction is an overreaching triple-jump of Olympic proportions. First, Sun asks that the inherently structural term "non-volatile storage" be interpreted as a means-plus-function limitation. Second, Sun asks that it be understood to include limitations from the specification that do not even pertain to "non-volatile storage." And third, Sun asks that these limitations be imported into the definition of "non-volatile storage" even though they are not required for either the invention or claimed function. NetApp's construction, by contrast, simply adheres to the well understood meaning of the term in the art, and should be adopted.

2. "meta-data for successive states of said file system"

Term	NetApp's Construction	Sun's Construction
"meta-data for successive states of said file system"	Information that describes successive "states of a file system" (as construed herein).	A block map file for recording snapshots of the file system.

The dispute between the parties is whether the term "meta-data for successive states of said file system" in claim 8 of the '292 patent should be construed to mean information that describes successive states of a file system, as NetApp proposes, or limited to a blockmap file for recording snapshots of the file system, as Sun proposes.

Sun's proposed construction should be rejected because it ignores the claim language and confuses a consistency point with a snapshot. As described throughout the specification, a new consistency point occurs whenever the root inode is updated to reflect a new set of self-consistent blocks, but a snapshot is a read-only copy of the file system at a particular point in time; in other words, not all consistency points are "recorded" as snapshots. *See generally* Ho Decl., Exh. A ('292 patent), cols. 11-23. In briefing the terms "consistent state" and "state of a file system," *see infra* at Section II.B.3, the parties agree that the two phrases are synonymous. Accordingly, the "successive states of said file system" referred to here are consistency points, not snapshots. It follows that, among other problems, Sun's proffered definition fails because it does not encompass the meta-data for any consistency points that are not preserved as snapshots. Moreover, even if "successive states" referred only to snapshots, which would be completely inconsistent with the intrinsic evidence, "meta-data" would not be limited to a block map file.

a. "Meta-Data For Successive States Of Said File System" Refers To Information Regarding Consistency Points, Not Snapshots

Sun correctly observes that asserted "[c]laim 8 pertains to creating read-only copies of successive consistent states of a file system." *See* Sun's Opening Cl. Constr. Br. at 10. Indeed, by doing so, Sun implicitly concedes that "successive states of said file system" refers to consistency points, not snapshots (*i.e.*, the copies). Yet, Sun completely ignores the language of the claim to argue that "meta-data for successive states of said file system" is limited to a block map file for recording snapshots. Meta-data is not only stored for snapshots, however, but for all consistency points. The claim language supports NetApp's plain meaning construction.

Independent claim 8 of the '292 patent recites:

> 8. A method for creating a plurality of read-only copies of a file system stored in blocks of a non-volatile storage means, said file system comprising meta-data identifying blocks of said non-volatile storage means used by said file system, comprising the steps of:
>
> storing **meta-data for successive states of said file system** in said non-volatile storage means;
>
> making a copy of said meta-data at each of a plurality of said states of said file system;

> for each of said copies of said meta-data at a respective state of said file system, marking said blocks of said non-volatile storage means identified in said meta-data as comprising a respective read-only copy of said file system.

See Ho Decl., Exh. A ('292 patent) at 26:1-15.

Put simply, the method of claim 8 has three steps: (1) storing meta-data for consistency points; (2) copying the meta-data at each consistency point for which snapshots are created;[4] and (3) marking which blocks correspond to which snapshot(s). NetApp's interpretation that "meta-data for successive states of said file system" refers to information that describes consistency points, not snapshots, is amply supported by the intrinsic evidence.

Nor is meta-data limited to "a block map file," as Sun argues. Indeed, the specification lists several examples of "meta-data," including an inode file, a root inode, a block map file, an inode map file, inode tables, directories, bitmaps, and indirect blocks. Ho Decl., Exh. A ('292 patent) at 9:18-10:56; 1:38-39. Moreover, it is commonly understood that "meta-data" includes many structures that contain information about the corresponding data. *See* Ganger Decl., ¶ 29. Unasserted claims can also be helpful to understanding claim 8. Indeed, the *en banc* decision in *Phillips* reversing the panel's claim construction turned on guidance from unasserted claims. As the Federal Circuit explained, "both asserted and unasserted, can be valuable sources of enlightenment as to the meaning of a claim term." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (*en banc*). Here, dependent claims 11, 12, 13, 18 and 19 are particularly instructive:

> 11. The method of claim 8 wherein said meta-data comprises pointers to a hierarchical tree of blocks comprising said file system.
>
> 12. The method of claim 8 wherein said meta-data comprises structures representing files of said file system.
>
> 13. The method of claim 12 wherein said structures representing files of said file system comprise inodes.
>
> 18. The method of claim 8 wherein said meta-data comprises a root structure referencing structures representing files of said file system, and

[4] It is worth noting that the copying step refers to "each of a **plurality** of said states" as opposed to "each successive state," which is consistent with the fact that not all consistency points are saved as snapshots.

> wherein said copies of said meta-data comprise copies of said root structure.
>
> 19. The method of claim 18 wherein said root structure comprises a root inode.

See Ho Decl., Exh. A ('292 patent) at 26:25-32; 26:50-55.

Because "independent claims are presumed to have broader scope than their dependents," the meta-data in claim 8 cannot be restricted to a block map file. *Acumed LLC v. Stryker Corp.*, 483 F.3d 800, 806 (Fed. Cir. 2007). The proper construction "meta-data for successive states of said file system" must therefore be at least broad enough to include pointers to a hierarchical tree of blocks, structures representing files of the file system (*e.g.*, inodes), and a root structure referencing structures representing files of said file system (*e.g.*, a root inode) because that is what the claims require. Sun's proposed construction, however, would not encompass any of the specific examples of meta-data recited in dependent claims 11, 12, 13, 18 or 19 and should be rejected.

b. Sun's Interpretation Should Be Rejected Due To The Doctrine Of Claim Differentiation

Under the doctrine of claim differentiation, "the presence of a dependent claim that adds a particular limitation gives rise to a presumption that the limitation in question is not present in the independent claim." *Phillips*, 415 F.3d at 1315 (citing *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 910 (Fed. Cir. 2004)). This doctrine, although only a guideline, is especially applicable when the parties dispute whether a limitation found in a dependent claim should be read into the independent claim, and the limitation in dispute is the only meaningful difference between the two claims. *Acumed*, 483 F.3d at 806.

Indeed, some of the unasserted dependent claims of the '292 patent have precisely the limitations Sun urges the Court to insert in the broad language of independent claim 8. Specifically, dependent claims 9 and 10 add the limitations "means for recording multiple usage bits per block" and "blockmap comprising multiple bit entries for each block," respectively:

> 9. The method of claim 8 wherein said step of marking said blocks comprising a respective read-only copy of said file system comprises placing an appropriate entry in a means for recording multiple usage bits

> per block of said non-volatile storage means.
>
> 10. The method of claim 9 wherein said means for recording multiple usage bits per block of said non-volatile storage means comprises a **blockmap** comprising multiple bit entries for each block.

See Ho Decl., Exh. A ('292 patent) at 26:16-24.

Because the "blockmap" limitation is the only meaningful difference between claims 9 and 10, there is a logical presumption that claim 9, from which claim 10 depends does not require a "block map comprising multiple bit entries for each block." Likewise, the additional step of "placing an appropriate entry in a means for recording multiple usage bits per block of said non-volatile storage means" is the only meaningful difference between claims 8 and 9. That limitation should therefore not be read into claim 8.[5] Sun's proposed construction, however, violates the doctrine of claim differentiation because it renders independent claim 8 identical in scope to dependent claims 9 and 10.

Although Sun argues in its brief that dependent claims 9 and 10 further define "the type of block map file to be used for recording snapshots," (*see* Sun's Opening Cl. Constr. Br. at 10-11) this argument is misplaced. First, claim 9 introduces an additional sub-step of placing an appropriate entry in a means for recording multiple usage bits per block, such as a block map. Claim 8, for the claim differentiation reasons described above, does not include this limitation. Second, claims 9 and 10 further modify the step of "marking said blocks," i.e., the third step recited in claim 8, as opposed to the "storing meta-data" step. Thus, there is no reason to believe, (and a person of skill in the art would not believe), that the reference to a block map in claim 10 is at all relevant to the proper construction of "meta-data for successive states of said file system." *See* Ganger Decl., ¶¶ 37-38.

c. Sun's Attempts To Import Limitations From The Specification And The '352 Patent Are Improper

The Federal Circuit has expressly recognized that a patentee's claims may embrace "different subject matter than is illustrated in the specific embodiments in the specification." *Nazomi Comm'ns, Inc. v. Arm Holdings, PLC*, 403 F.3d 1364, 1369 (Fed. Cir. 2005). Even if the

---

[5] Although it does not depend from claim 8, claim 3 also expressly recites "blockmap file data," demonstrating that the absence of a "blockmap" limitation in claim 8 is significant.

patent describes only a single embodiment, the claims of the patent are *not* normally construed as being limited to that embodiment "because persons of ordinary skill in the art rarely would confine their definitions of terms to the exact representations depicted in the embodiments." *Phillips*, 415 F.3d at 1323. Indeed, a person of ordinary skill reading the '292 patent would recognize that the WAFL system described therein is merely a preferred embodiment. *See* Ganger Decl., ¶ 39. There are various other ways that a file system could keep track of which blocks correspond to which snapshot(s) besides using a block map. *Id.* at ¶ 36.

The phrase "the present invention" does not necessarily restrict the scope of the invention to the preferred embodiment either. *See Karlin Tech. Inc. v. Surgical Dynamics, Inc.*, 177 F.3d 968, 973 (Fed. Cir. 1999) (noting that there were mixed references to "present invention" and "preferred embodiment" in the specification); *MGP Ingredients, Inc. v. Mars, Inc.*, 494 F.Supp.2d 1231, 1237 (D. Kans. 2007) (finding there was no clear disavowal of the full scope of the claim language despite repeated references to "the present invention"); *Boston Scientific SciMed, Inc. v. ev3 Inc.*, 502 F.Supp.2d 931 (D. Minn. 2007) (considering the totality of the intrinsic evidence, including a claim-differentiation presumption, to find that the disputed phrase was not limited by a characterization of "the present invention" in the specification).

Here, the use of the phrase "the present invention" does not restrict the claims of the '292 patent to the preferred method of keeping track of which blocks correspond to which snapshot(s). First, the specification explicitly states that it describes numerous features in detail but "the present invention ***may be practiced without these specific details***." *Id.* at 5:37-42. Second, as discussed above, the '292 patent includes numerous dependent claims that are directed to particular features of the preferred embodiment. Claim 8 itself has 12 dependent claims, including claim 10 which expressly recites a "blockmap" limitation. If the patentee had intended to limit all the claims of the '292 patent to require the use of a block map file by saying "the present invention," as Sun argues, it would not have been necessary to refer to a "blockmap" in claim 10 because it would have already been implied. Third, the Summary of the Invention describes three distinct functions of the preferred WAFL file system, namely: (1) maintaining a file system in a consistent state; (2) creating snapshots; and (3) ensuring that new data would not

overwrite old data that was being saved in a snapshot. Significantly, there are different independent claims of the ’292 patent directed to each of these functions, *see, e.g.*, claims 1, 4 and 8, which suggests that the examiner agreed there were at least three distinct aspects of the invention that were independently patentable. In short, Sun improperly attempts to limit the claims of the ’292 patent to one particular implementation of the preferred embodiment.

Regardless, none of the evidence referenced by Sun in the specification or the ’352 patent is sufficient to overcome the strong presumption that the phrase “meta-data for successive states of said file system” in claim 8 is not limited to a block map file. Even in the preferred embodiment, the “meta-data” that is copied to make a new snapshot is a root inode, not a block map file. *See, e.g.*, Ho Decl., Exh. A (’292 patent) at 18:13-16; 18:65-19:6. Moreover, while the preferred embodiment utilizes a block map file to keep track of whether each block is part of the active file system and/or one or more snapshots, it is updated, *but not copied*, whenever a new snapshot is made. *Id.* at 18:19-23; Ganger Decl., ¶ 41. In other words, the block map file is not the “said meta-data” that is copied in the second step of claim 8.

Because Sun’s interpretation of this claim term is wholly inconsistent with both the claim language and the teachings of the specification, its attempt to import a “block map” limitation into claim 8 should be rejected.

3. “file system information structure”

Term	NetApp’s Construction	Sun’s Construction
“file system information structure”	A data structure containing information about the layout of a file system.	Data structure that contains the root inode of a file system in a fixed location on disk.

The dispute between the parties is whether the term “file system information structure” in claim 4 of the ’292 patent should be given the meaning suggested by the claim language itself, as NetApp proposes. Sun’s attempt to import additional limitations from the preferred embodiment should be rejected, particularly in light of compelling intrinsic evidence that the scope of the invention was not limited in this way.

a. NetApp's Proposed Construction Is Rooted In The Language Of The Claims

Independent claim 4 of the '292 patent recites:

> 4. A method for maintaining a file system comprising blocks of data stored in blocks of a non-volatile storage means at successive consistency points comprising the steps of:
>
> storing a first **file system information structure** for a first consistency point in said non-volatile storage means, said first **file system information structure** comprising data describing a layout of said file system at said first consistency point of said file system;
>
> writing blocks of data of said file system that have been modified from said first consistency point as of the commencement of a second consistency point to free blocks of said non-volatile storage means;
>
> storing in said non-volatile storage means a second **file system information structure** for said second consistency point, said second **file system information structure** comprising data describing a layout [of] said file system at said second consistency point of said file system.

See Ho Decl., Exh. A ('292 patent) at 25:12-29.

As explicitly recited in claim 4, the "file system information structure" comprises ***data describing a layout of the file system*** at either a first or second consistency point of the file system. Sun's objection to the inclusion of this phrase in the claim construction is therefore meritless because the claim language itself requires it. NetApp's proposed inclusion of the phrase "information about the layout of a file system" should be adopted because it is drawn directly from the language of the claim itself.[6] Moreover, a person of skill in the art would understand that, while the '292 patent contemplates that other types of miscellaneous information about the file system (*e.g.*, "the number of blocks in the file system, the creation time of the file system," and a checksum, *see* Ho Decl., Exh. A ('292 patent) at 10:65-11:1) might also be part of a commercial file system information structure, information about the layout of the file system at a given consistency point would certainly be included in the file system information structure as reflected by the claim language. *See* Ganger Decl. ¶ 45.

---

[6] Sun's argument that NetApp's construction creates a tautology is misplaced. Claim 4 recites "storing a first file system information structure;" if "a file system information structure" was defined to mean a data structure that is stored, **that** would create a tautology.

The *Mangosoft* case cited by Sun (*see* Sun's Opening Cl. Constr. Br. at 17) is distinguishable because the rejected construction of the disputed claim term simply restated an implied relationship between other components of the system. *Mangosoft, Inc. v. Oracle Corp.*, 525 F.3d 1327, 1330-31 (Fed. Cir. 2008). In contrast, "comprising data describing a layout of said file system" expressly modifies and defines what is contained in "said file system information structure." In addition, the *Mangosoft* court concluded that a "local" persistent memory device was distinct from a "shared," "networked," or "remote" device because the patentee had added the word "local" during prosecution to distinguish prior art. *Id.* at 1331. There was no such disclaimer during prosecution of the '292 patent.

b. The Doctrine Of Claim Differentiation Compels An Interpretation Of "File System Information Structure" Not Limited To a Data Structure Stored In a Fixed Location

As set forth above, independent claim 4 requires, *inter alia*, "storing a first file system information structure for a first consistency point in said non-volatile storage means" and "storing in said non-volatile storage means a second file system information structure." Dependent claim 5 recites additional limitations for both storing steps in claim 4, namely (1) "storing first and second copies of said first file system information structure **at first and second locations** respectively of said non-volatile storage means" and (2) overwriting the first and second copies of the first file system information structure with the first and second copies, respectively, of the second file system information structure. Dependent claim 6 further recites:

> 6. The method of claim 5 wherein **said first and second locations of said non-volatile storage means comprise fixed predetermined locations** of said non-volatile storage means.

As explained above, the doctrine of claim differentiation raises a presumption that a limitation set forth in a dependent claim should not be read into the independent claim. Since the only meaningful difference between claims 5 and 6 of the '292 patent is that claim 6 adds the limitation of storing copies of the file system information structure in "fixed predetermined locations" of the non-volatile storage means, there is a strong presumption that neither claim 5 nor asserted claim 4, from which claim 6 depends, requires storing the file system information structure to a fixed location on disk.

Notably, the case cited by Sun in support of the proposition that a term should not be construed to encompass a second limitation, (*see* Sun's Opening Cl. Constr. Br. at 17) applies the doctrine of claim differentiation. *AllVoice Computing PLC v. Nuance Communications, Inc.*, 504 F.3d 1236-1247-48 (Fed. Cir. 2007) (refusing to construe "forming" in one independent claim to include "updating" when other independent claims in the patent recited both functions). Here, Sun's proposed construction of "file system information structure" should be rejected because it would render the "fixed predetermined locations" limitation of claim 6 superfluous.

c. Sun Improperly Imports Limitations From The Specification

Sun's proposed construction violates the well-established axiom that claims should not be limited by the embodiment(s) described in the specification absent good reasons to do so. *See Phillips*, 415 F.3d at 1323; *Liebel-Flarsheim Co., v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004) ("Even when the specification describes only a single embodiment, the claims of the patent will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using 'words or expressions of manifest exclusion or restriction.'") (internal citation omitted). Specifically, Sun improperly attempts to import two additional limitations from the preferred embodiment: (1) that the file system information structure must be stored in a fixed location on disk[7] and (2) that the file system information structure must contain the root inode.

As discussed above, the file system information structure should not be limited to being stored in a fixed location on disk. The specification also expressly teaches that the file system information structure does not necessarily contain a root inode:

> "Snapshots are created by duplicating the root data structure of the file system. In the preferred embodiment, the root data structure is the root inode. However, **any data structure representative of an entire file system could be used**."

See Ho Decl., Exh. A ('292 patent) at 18:13-16. Thus, while the preferred embodiment contemplates that the fsinfo block includes a root inode, *id.* at 9:33-35, it is clear that **any data structure representative of an entire file system** could be stored in the fsinfo block in lieu of a

[7] Incidentally, Sun's argument that the file system information structure is stored in a fixed location **on disk** is inconsistent with its position that "non-volatile storage means" lacks sufficient structure to avoid being construed under § 112, ¶ 6.

root inode. Indeed, a person of skill in the art would understand from this passage that the invention does not require the file system information structure to include a root inode. *See* Ganger Decl., ¶ 48.

d. Sun Conflates The File System Information Structure With The File System Information Block

In a futile attempt to distract the Court with a red herring, Sun repeatedly argues that the specification equates a “file system information (fsinfo) structure” with a “file system information (fsinfo) block.” This is inaccurate. First, the use of the parenthetical abbreviation for “file system information” in both has no bearing on the proper interpretation of a “fsinfo **structure**” versus “fsinfo **block**.” Second, “file system information block” has two usages in the specification. *See* Ganger Decl., ¶ 50. The first meaning of “fsinfo block” is a location on disk. *See, e.g.*, Ho Decl., Exh. A (’292 patent) at 9:33-35 (describing in the preferred embodiment, “a fixed location on disk referred to as the file system information (fsinfo) block described below”). The second meaning of “fsinfo block” refers to the data structure itself, as opposed to where it is stored. *See, e.g.*, *id.* at 13:66-67 (“The fsinfo block is written twice. It is first written to one location and then to a second location.”) In the latter usage, “fsinfo block” is synonymous with “fsinfo structure.” It would be clear to a person of skill in the art reading the ’292 patent which meaning of “fsinfo block” applies, depending on the context of different passages in the specification. *See* Ganger Decl., ¶ 51. Third, and most importantly, while “file system information block” has two meanings in the specification, “file system information structure” only has one. *Id.* at ¶ 52. Indeed, “file system information structure” is used consistently throughout the specification to only refer to the data structure itself, not its location, much less a fixed location. *Id.*

Significantly, none of the claims in the ’292 patent recite a “file system information block,” so the alternate meaning of that phrase is irrelevant. The problem with Sun’s proffered construction is that it focuses almost solely on the “location” definition of “file system information block” in lieu of the definition that would be synonymous with “file system information structure.”

For the reasons above, Sun's naked attempts to import limitations from the specification that are inconsistent with language of the claims should be rejected.

B. U.S. PATENT NO. 6,892,211

1. Pointing directly and indirectly to buffers in said memory and a second set of blocks on said storage system" (Claims 1, 9, 17)

The table below shows the parties' respective positions for these terms.

Term	NetApp's Construction	Sun's Construction
"Pointing directly and indirectly to buffers in said memory and a second set of blocks on said storage system"	No construction necessary, plain and ordinary meaning. To the extent the Court deems a construction necessary, the term means, "Pointing directly to blocks and/or buffers, and/or indirectly to blocks and/or buffers."	"Pointing directly and indirectly to buffers in said memory and pointing directly and indirectly to a second set of blocks on said storage system."

The dispute between the parties is whether the incore root inode can point to blocks and buffers using any combination of direct and indirect pointers (NetApp's position) or whether it must point both directly and indirectly both to blocks and buffers (Sun's position). NetApp's primary position – that the plain and ordinary meaning is sufficient – comports with the claim language and the specification. To the extent that NetApp proposes a construction, NetApp aims to clarify that the '211 patent discloses a group of pointers (*i.e.*, direct and/or indirect pointers) which collectively point to a group of data locations (*i.e.*, buffers and blocks). Sun's construction, however, is little more than an attempt to enforce a non-existent rule of grammar. Simply put, Sun wants the Court to believe that this dispute is over the meaning of the word "and".[8]

Sun makes the arguments about this term extremely complicated in order to make the disputed term seem like something other than what it is. But the underlying concept is

[8] It is especially ironic that Sun is now attempting to create a dispute out of an imaginary rule about the meaning of the word "and", urging the Court that "and" cannot mean "or". Sun has served at least nine sets of written discovery on NetApp since the commencement of litigation between the parties that contain some form of the instruction that, "'and' and 'or' shall be construed conjunctively or disjunctively, whichever makes the request more inclusive." Ho Decl., Exh. F (excerpts from Sun's discovery requests). As shown here, Sun plainly understands that in the context of legal drafting "and" is perfectly amenable to including "or" in its range of possible meanings depending on context.

actually simple. There must be an incore root inode that points to the data making up the second consistent state of the file system. Ganger Decl., ¶ 58. The disputed limitation requires that there be some locations for the computer data in question (*i.e.*, buffers and blocks—either will do, both is normal) and pointers that point to them (direct or indirect—either will do, but again, both is normal). As long as this condition is met, the disputed term is satisfied. That is, as long as this condition is met, the incore root inode points to the data that makes up the second consistent state of the file system.

The disputed term recites an incore root inode that points directly and indirectly to a *group* of things. Sun and NetApp agree that the claim does not require the incore root inode to point directly and indirectly to a given block or buffer at the *same* time. Sun's Opening Cl. Constr. Br. at 21. Because Sun does not contend that the incore root inode must point in both ways to *each individual block or buffer*, Sun must concede that the first "and" can contextually mean "and/or". Nevertheless, Sun contends that according to the strictures of grammar (uncited) the second "and" can only possibly mean "both". Sun argues in its opening brief that NetApp is misusing the word "and" by translating it to mean "and/or". *See* Sun's Opening Cl. Constr. Br. at 21. This is ironic in light of the fact that NetApp understands that "and" must be understood contextually, whereas Sun attempts to demand that it can only mean what Sun says it means—even while that meaning changes on a case-by-case basis even within the same claim.

a. The Plain And Ordinary Meaning Of The Claim Language Supports NetApp's Position

The structure of the claim – particularly the language surrounding the disputed term – supports NetApp's position that there is no requirement for both direct and indirect pointers to both buffers and blocks. Indeed the plain language of the claim establishes that the group "blocks and buffers" is to be understood as a whole. The relevant portion of claim 9, reformatted for readability, is as follows:

> [W]herein said memory also stores information including instructions executable by said processor to maintain said file system, the instructions including the steps of

> (a) maintaining an on-disk root inode on said storage system, said *on-disk root inode pointing directly and indirectly to a* ***first set of blocks*** *on said storage system* ***that store a first consistent state of said file system***, and
>
> (b) maintaining an incore root inode in said memory, said *incore root inode pointing directly and indirectly to* ***buffers in said memory and a second set of blocks*** *on said storage system,* ***said buffers and said second set of blocks storing data and metadata for a second consistent state of said file system***, said second set of blocks including at least some blocks in said first set of blocks, with changes between said first consistent state and said second consistent state being stored in said buffers and in ones of said second set of blocks not pointed to by said on-disk inode.

Ho Decl., Exh. B ('211 patent) at 24:45-62. As the emphasized portions of the above quotation show, there is a parallel structure in this claim in which the first set of blocks (limitation "(a)") is the group that represents the first consistent state of the file system and the second set of blocks, together with buffers (limitation "(b)"), is the group that represents the second consistent state of the file system. The on-disk root inode and the incore root inode anchor the first and second consistent states of the file system respectively. Each of these groups is therefore meant to be treated as a whole—and is defined as the effective domain of its respective type of root inode. Thus, there is no requirement that the subparts of the second group (*i.e.*, the blocks and buffers) should be treated separately from each other.

Indeed, Claim 9 recites, "said buffers and said second set of blocks storing data and metadata for a second consistent state of said file system". The first appearance of the word "and" here is necessarily being used to establish a single combined group, because the claim describes the "buffers and second set of blocks" together as that which represents the second consistent state of the file system. Ho Decl., Exh. B ('211 patent) at 24:55-57. Because the subsequent use of the same phrase makes it clear that the "buffers and second set of blocks" are meant to be treated as a single group, there is no arbitrary requirement imposed that the two be thought of separately for the purpose of applying each adverb—"directly and indirectly"—separately to each. *See* Ganger Decl. ¶ 62. Thus, the plain and ordinary meaning of the claim language supports NetApp's position that the claim requires an incore root inode pointing directly and/or indirectly to blocks and/or buffers taken together as a whole group representing a second consistent state of a file system.

b. The Specification Supports NetApp's Position

NetApp's position is simple: as long as the incore root inode has direct and/or indirect pointers pointing to the group of blocks and buffers composing the second state of the file system, the grammar of the term is satisfied. In other words, there is no requirement that the incore root inode point both directly and indirectly to blocks *and* point both directly and indirectly to buffers. The specification supports this position by describing a preferred embodiment in which the incore root incore does not always point directly *and* indirectly to blocks and point directly and indirectly to buffers. Indeed, there is no requirement that the incore root inode have both direct and indirect pointers at all, but simply that it point directly and/or indirectly to the buffers and/or blocks that store a second consistent state of the file system. For instance, a file system as described in the specification *can* be in a state where the incore root inode points to all the buffers directly; it describes another state where the incore root inode points only indirectly to blocks. *See* Ganger Decl. ¶¶ 64-65 (describing how instances of the incore root inode not pointing directly to any blocks, and not pointing indirectly to any buffers as enabled by the specification of the '211 patent itself).

This means that the specification supports NetApp's position that there is no requirement for the incore root inode to point both directly and indirectly to buffers and to point directly and indirectly to blocks on disk. This alone demonstrates that Sun's proposed construction, which requires that there always be both direct and indirect pointing to buffers, and direct and indirect pointing to blocks, must be wrong. More importantly, it confirms that the grammar of the claim is satisfied if, at a minimum, there are direct and/or indirect pointers and they point to blocks and/or buffers collectively as the group of data representing a second consistent state of the file system.

The Federal Circuit has recently ruled against the position now espoused by Sun in holding that this reading of "and" is *not* required, where the context suggests that it is meant to provide *options* disjunctively rather than *requirements* conjunctively:

> The claim also does not use *and* in isolation but in a larger context that clarifies its meaning. Specifically, *and* appears in conjunction with the adverbs *independently* and *together*. **As the district court explained,**

> **these terms signal that *and* links alternatives that occur under the different conditions of independence or togetherness**.

Ortho-McNeil Pharmaceutical, Inc. v. Mylan Labs., Inc., 520 F.3d 1358 at 1362 (Fed. Cir. 2008) (italics in original, bold added). Thus the Federal Circuit has specifically held that, where context suggests it is the more reasonable alternative, "and" should be read to denote options rather than requirements. In light of the above analysis of the claims and the specification, Sun cannot now arbitrarily impose a restrictive meaning on the word "and".

c. Sun's Construction Is Wrong

As explained above, Sun's argument is simply a misguided attempt to translate the claim term into a kind of algebraic expression where direct and indirect must be distributed over blocks and buffers each alone. As explained above, this approach simply does not account for the true meaning of the disputed term because it arbitrarily breaks up the group of buffers and blocks, requiring each of these things to individually meet all the requirements that were only intended to apply to the group as a whole. In other words, Sun wishes to apply the "and" in "directly and indirectly" to buffers and blocks as separate requirements.

But, as explained above, the specification actually discloses at least three possible states of a preferred embodiment in which only one of the two possible types of pointers is being used for a given group of buffers, blocks, or both. Sun's proposed construction – demanding both direct and indirect pointers at all times – must thus be wrong. And, one of ordinary skill in the art would understand that an incore root inode must be able to point to blocks on disk to the extent the file system itself was on disk. In other words, if the files stored by the file system are on disk, then ultimately the root inode, whether incore or on disk, needs a way of pointing to them. With this in mind, there are only two ways it can do so: directly and indirectly. While there is certainly no reason an incore root inode could not point directly to blocks on disk – and indeed a preferred embodiment does just that – it is completely possible for an incore root inode to use only indirect pointers to point to blocks on disk. *See* Ganger Decl. ¶¶ 64-65, *supra*, at 25. Nothing about this arrangement would in any way defeat the inventiveness of the '211 patent, and one of ordinary

skill in the art would recognize this immediately. Yet Sun's proposed construction demands that the root inode point directly to on-disk blocks.

Even if one sets aside the teachings of the specification, and considers nothing but the grammar of the disputed claim term, there is simply no reason to prefer a narrowing algebraic interpretation of this grammar to require that direct and indirect must apply to each buffers and blocks separately. For example, there is no dispute that pointing directly requires a direct pointer and pointing indirectly requires an indirect pointer. Just as in the '211 patent, the grammar is not meant to speak to this issue. The limitation is satisfied as long as there are direct and indirect pointers, pointing to the group comprising blocks or buffers that represent a second consistent state of the claimed file system.

For these reasons, "pointing directly and indirectly to buffers in said memory and a second set of blocks on said storage system," means, "Pointing directly to blocks and/or buffers, and/or indirectly to blocks and/or buffers."

2. "root inode"

Term	NetApp's Construction	Sun's Construction
"root inode"	An inode that points directly and/or indirectly to all the blocks in a consistent state of a "file system" (as construed herein).[9]	The index node data structure stored in a fixed location that roots a set of self-consistent blocks on the storage system that comprise the file system.

As correctly noted by Sun, the fundamental dispute between the parties—similar to the parties' dispute with respect to "file system information structure"—is whether a "root inode" must be stored in a fixed location. NetApp incorporates by reference its explanation of why the Court should not construe "file system information structure" to mean a data structure that contains the root inode of a file system in a fixed location on disk. *See* Section II.A.3. Further, Sun's attempts to read in limitations of the preferred embodiment should be rejected.

[9] Sun complains that NetApp's constructions include "as construed herein" references to other terms that are not currently being construed by the Court. This argument should be accorded no weight because the parties' proposed constructions are taken from the charts in the Joint Claim Construction Statement which was submitted before the Court decided to limit the briefing to 14 terms.

a. The Claim Language Itself Compels An Interpretation Of "Root Inode" That Is Not At A Fixed Location

The claims of the '211 patent refer to both an "on-disk root inode" and an "incore root inode." Thus, the phrase "root inode" by itself must be construed in a manner sufficiently broad to accommodate both. Because Sun's proffered construction would be limited to a specific embodiment of an on-disk root inode, it should be rejected.

The parties agree that if "incore" were construed, it would mean "in memory" as opposed to on disk. *See* Ho Decl., Exh. G (Joint Cl. Constr. and Prehearing Statement, Exh. B) at 4. All of the references in the specification cited by Sun refer to storing the root inode in a fixed location **on disk** in the preferred embodiment. These passages in no way limit the meaning of "incore root inode," much less "root inode" with no modifier. In fact, in the preferred embodiment, the incore inodes, including the incore root inode, are larger than the corresponding on-disk inodes and contain contents that are never written to disk, much less in a fixed location. *See* Ho Decl., Exh. B ('211 patent) at Fig. 8 and accompanying text (showing an incore inode that includes incore information, a copy of the on-disk inode, a WAFL buffer data structure, and 16 buffer pointers).

Moreover, a person of skill in the art reading the '211 patent would understand that an "incore root inode," like other information temporarily stored in memory, is often kept in whatever available location in memory is allocated by the memory management system. *See* Ganger Decl. ¶¶ 75-76. The specification contains nothing suggesting that an "incore root inode" is assigned a location in memory differently from any other information temporarily stored in memory, much less in a fixed location. *Id.* at ¶ 77. Since incore inodes are not stored in a fixed location in memory, a "root inode" should not be construed to require that it be stored in a fixed location because, among other reasons, it would render the claim term "incore root inode" nonsensical.

b. Even An "On-Disk Root Inode" Is Not Necessarily Stored At a Fixed Location

Even if Sun's arguments were solely directed to an "on-disk root inode," the referenced passages in the specification do not require an on-disk root inode to be stored at a

fixed location. Because the ’211 patent is a continuation of, and shares the same specification as, the ’292 patent, all of the arguments why a file system information structure need not be stored in a fixed location on disk, *supra* Section II.A.3, apply equally here. As Sun correctly points out, the preferred embodiment describes an on-disk root inode stored in the file system information block. Given that, the on-disk root inode cannot be required to be stored in a fixed location for the same reason that the file system information structure need not be stored in a fixed location.

Sun agrees that the specification teaches that a root inode is an inode that roots a set of self-consistent blocks. *See* Sun’s Opening Cl. Constr. Br. at 25. In other words, a root inode points directly and/or indirectly to all the blocks in a consistent state of a file system. It would be inappropriate to read in a “fixed location” limitation that is “wholly apart from any need to interpret what the patentee meant by particular words or phrases in the claim.” *nCube Corp. v. SeaChange Int’l, Inc.*, 436 F.3d 1317, 1322 (Fed. Cir. 2006) (*citing E.I. du Pont de Nemours & Co. v. Phillips Petroleum Co.*, 849 F.2d 1430, 1433 (Fed. Cir. 1988)). The thrust of Sun’s argument appears to be that the root inode of the active file system must be stored at a fixed location so it can be located during booting of the file system. Yet, one of skill in the art would understand that there are other mechanisms for ensuring that the root inode for the current consistency point can always be located. *See* Ganger Decl. ¶ 79. For example, rather than storing the root inode in a fixed location, the file system could simply store a pointer to the root inode in a fixed location. *Id.* Indeed, at the precise moment that a new snapshot is created, at least one copy of the root inode for the active file system is **not** stored in a fixed location. *See id.* at 17:61-63; 18:17-19; 18:57-67; 19:5-8; Ganger Decl. ¶ 80.

In any event, the specification clearly indicates—and one of ordinary skill in the art would understand—that copies of the root inodes for previously saved consistency points (*i.e.*, snapshot inodes) can be stored anywhere in the file system. Ganger Decl. at ¶ 81. First, the fact that more than one snapshot can be stored necessarily means that the root inode for the file system at a given consistency point is not always copied to the same location in the inode file or it would overwrite a previous snapshot. *See* Ho Decl., Exh. B (’211 patent) at 18:1-8; Fig. 22 and accompanying text. Second, while the specification states that “[t]he snapshot inodes reside in a

fixed location in the inode file," *id.* at 18:9-10, the inode file (like any other file) can be stored anywhere in the file system. *Id.* at 9:19-29; 18:58-63. Thus, the specification expressly contemplates at least some root inodes are not stored in fixed locations on disk.

For all of the aforementioned reasons, the Court should reject Sun's proffered construction and adopt NetApp's definition of "root inode."

3. "consistent state"/"state of a file system"[10]

Term	NetApp's Construction	Sun's Construction
"consistent state"/"state of a filesystem"	A set of blocks on disk, rooted by a root inode, that includes all the blocks required for the data and file structure of a file system.	A set of storage blocks for that file system that includes all blocks required for the data and structure of the file system.

The dispute between the parties is whether a "consistent state" or a "state of a file system" must be rooted by a "root inode" (NetApp's position) or not (Sun's position).

Sun asserts in its brief that the "language in the '211 specification that supports [its] construction of 'consistent state' also supports [its] construction of 'state of a file system' in the '292 patent." *See* Sun's Opening Cl. Constr. Br. at 19 n.5. But Sun does not actually point to any evidence from the common specification shared by these patents to support its construction. Instead, Sun relies on U.S. Patent No. 7,174,352, which was filed roughly eight years after '292 and '211 specifications were drafted and which does not even include all the same inventors as the '292 and '211 patents. Sun's heavy reliance on this later-filed patent is, at first blush, odd, because in the Joint Claim Construction and Prehearing Statement, Sun cited to a great deal of evidence from the '292 and '211 patents that supposedly supported its proposed construction. A closer look at Sun's evidence from the Joint Statement, however, reveals that Sun likely chose to ignore it because it confirms that a "state of a file system" or a "consistent state" is, in fact, "rooted by a root inode," just as NetApp proposes.

For instance, in the Joint Claim Construction and Prehearing Statement, Sun cites to three passages in the specification of the '292 patent that summarize the "states" a file system

[10] The term "consistent state" appears in the '211 patent, while the term "state of a file system" appears in the '292 patent. Because the '211 patent is a continuation of the '292 patent, the two patents share the same specification. The parties agree that the construction of these two claim terms should be the same.

can take on – describing them as either "consistent states" or "consistency points" – and that explicitly note how these things are "rooted by a root inode":

> Changes to the file system are tightly controlled to maintain the file system in a consistent state. The file system progresses from one self-consistent state to another self-consistent state. The set of self-consistent blocks on disk that is *rooted by the root inode* is referred to as a consistency point.

See Ho Decl., Exh. D (citing the '292 patent) at 4:9-13; *see also id.* at 11:60-12:8 (offering a nearly identical description) *id.* at Abstract (same). Other evidence Sun cited to is in complete accord with this description:

- In the Joint Claim Construction and Prehearing Statement, Sun cites to Figure 16 and the accompanying description of a "file system" in a "consistent state." This unambiguously shows the "consistent state" rooted by a root inode. *See id.* at 11:6-27; *id.* at Fig. 16.
- Sun cites to Figures 17A-17L and their accompanying description, which describe the generation of a "consistency point." These figures again depict the "consistent state" of the file system as being anchored by a root inode, just as NetApp proposes. *See id.* at 15:29-17:64; *id.* at 22:25-23:18; *id.* at Figs. 17A-17L.
- Sun cites to Figures 20A-C and their accompanying description. These figures explicitly depict two copies of the "root inode" pointing to, and thus anchoring, a generic "file system," which the text explains are "consistency points." *See id.* at Figs. 20A-C and accompanying description.

Thus, the bulk of the evidence Sun points to in support of its proposed construction confirms that a "state of a file system" or a "consistent state" is "rooted by a root inode."

Notably, Sun's proposed construction for the term "consistency point" requires that a "consistency point" be "rooted by a root inode." *See* Ho. Decl., Exh. D (Supp. Joint Cl. Constr. and Prehearing Statement), Exh. A. Sun must concede that this term should be construed like the term "consistent state." Indeed, there is great overlap in the evidence Sun cites in support of its proposed constructions for "consistent state" and "consistency point." Yet Sun's brief offers no explanation as to why a "consistency point" should be "rooted by a root inode" while a "consistent state" should not. NetApp, on the other hand, proposes uniform constructions for the terms "state of a file system," "consistency point," and "consistent state" all reflecting that these things are "rooted by a root inode." NetApp's constructions should thus be adopted.

C. U.S. PATENT NO. 7,200,715

1. Background

The ’715 patent is generally directed to a technology for improving the read and write efficiency of a RAID array in combination with a file system. RAID, which stands for “**R**edundant **A**rray of **I**nexpensive **D**isks”, is a technique whereby a number of disks are used together for storing data in an arrangement that allows the contents of any given failed disk to be recreated as a function of the data on all the other disks. The techniques of RAID generally allow data to be protected even when individual disks storing data experience failure. This is accomplished by a RAID architectural concept called a “stripe” defined as a grouping of space on multiple disks including a uniform amount of space on each of a number of disks, at least one of which is used to store parity (also called “redundancy data”) and the others of which are used to store data. In the event that one of the disks should fail, the data that is lost when the disk fails can be reconstructed by performing a mathematical function such as arithmetic modulo two on the remaining segments of data and parity in the same stripe. A standard form of RAID includes one block per disk in a stripe and one parity block to protect the others.[11] The cost effectiveness and increased reliability of such a system is tremendous. *See generally* Ho Decl., Exh C. (’715 patent); Ganger Decl., ¶¶ 83-88.

There is a conflict between optimizing the functionality of file systems and individual disks on the one hand and optimizing the performance of a RAID on the other. File systems and disks are generally optimized to write large amounts of related data contiguously on an individual disk. RAIDs, on the other hand, tend to be optimized to write a full stripe all at once in order to minimize the amount of time spent reading data from disks in order to recalculate parity. Indeed, if an entire stripe is written at once, all the blocks in the stripe are in memory, and parity can be calculated without needing to read anything from any of the remaining disks. The result of this type of optimization is that related data tends to be broken up into small pieces and

[11] This standard type provides protection in the event that one disk in the RAID should fail. In this type of implementation each block is part of exactly one stripe. However there are other implementations. For instance each block can be a member of two stripes rather than one, each protected by a different parity block, thereby providing for protection even if two disks in the RAID should fail. Ganger Decl., ¶ 85.

written on a number of disks at once. If this is done for a number of different write requests, then each time any one of the files containing those small pieces needs to be accessed, each disk in the RAID has to move its data access "head." Eventually, performance suffers because the time spent moving heads becomes large relative to the time actually spent reading and writing data. *See generally*, Ho Decl., Exh C. ('715 patent); Ganger Decl. ¶ 86.

Ideally, a file system with a RAID would be able to simultaneously accomplish both disk I/O optimization *and* parity recalculation optimization. The inventors of the '715 patent disclose a way of doing that. Specifically, the '715 patent discloses a data structure, called an "association," which aggregates multiple write requests and assigns related data to contiguous locations on a *single* disk. The multiple requests are ultimately written on multiple disks in the RAID in order to write whole stripes at a time. The result of this optimization is that the file system usually commands the RAID layer to write more than one stripe at the same time. The file system can do this because it has a map of the disk architecture, called "topology information," that it uses to assign individual blocks of data to the locations on the several disks of the RAID where they can be written most efficiently. Because related data will tend to be written on the same disk, and full stripes will tend to be written at once, this idea enables RAID *and* file system/disk optimization at the same time. *See generally*, Ho Decl., Exh C. ('715 patent); Ganger Decl. at ¶ 87.[12]

2. "Associating the data blocks with one or more storage blocks across the plurality of stripes as an association," (Claims 21 and 52), and "The association to associate the data blocks with one or more storage blocks across the plurality of stripes" (Claim 39)

The table below shows the parties' respective positions for these terms.

Term	NetApp's Construction	Sun's Construction
"associating the data blocks with one or more storage blocks across the plurality of stripes as an association"	No construction necessary, plain and ordinary meaning. To the extent the Court deems a construction necessary, the term means, "creating a data	Sun contends this phrase is indefinite under 112, ¶ 2. If not indefinite: "Associating each data block with a respective one of the storage blocks across the plurality of stripes as an

[12] In one embodiment disclosed in the '715 patent it is apparent that the storage blocks are the same size as the data blocks. Of course this is not necessary and in many conventional RAID schemes it will not be the case. This is in no way essential to the core concepts revealed by this patent. *See* Ganger Decl. ¶ 88.

	structure that relates data blocks to locations on more than one stripe."	association."
"the association to associate the data blocks with one or more storage blocks across the plurality of stripes"	No construction necessary, plain and ordinary meaning. To the extent the Court deems a construction necessary, the term means, "the data structure relating data blocks to locations on more than one stripe."	

The dispute between the parties is whether these terms recite a data structure capable of mapping one or more data blocks to locations on more than one of many stripes of a storage array (NetApp's position) or whether the text of the asserted claims should be rewritten to import a limitation from other claims that were not asserted in this case (Sun's position).

a. The Plain And Ordinary Meaning Of The Words And Grammar In The Disputed Terms Demonstrates What Is Meant

No construction is necessary because the plain and ordinary meaning of the claim terms effectively conveys their meaning. Claim 21, for example, requires "*associating* the *data blocks* with one or more *storage blocks* across the plurality of stripes as an association." Ho Decl., Exh C. ('715 patent) at 21:47-49. One of ordinary skill reading this language would understand that there are two different types of blocks at issue—data blocks and storage blocks — and that blocks of the former type are being associated with blocks of the latter type. *See* Ganger Decl. ¶ 97. The phrase "as an association," by its use of the noun "association," makes clear that the "association" is meant to be a thing in itself. Specifically, one of ordinary skill would understand that the "association" is a data structure whose purpose is to maintain and convey the assignment of certain data blocks to be stored at certain storage blocks. *See id.* at ¶ 93. Claim 1, which includes the limitation "transmitting the association to a storage device manager…," confirms this. Ho Decl., Exh C. ('715 patent) at 20:10. Indeed, this limitation demonstrates that the '715 inventors claimed the "association" as a special data structure for the purpose of storing the relationship between data blocks and the storage blocks where they are to be written. If the inventors had intended otherwise, they would have simply claimed "transmitting the data blocks."

Additionally, a person of ordinary skill would understand that the phrase "one or more" means that sometimes only one storage block is used by a given "association." Because the specification already defines "storage block" in the context of the '715 patent to be the portion of a stripe that is located on an individual disk, one of ordinary skill in the art would understand that a single storage block will not exist on multiple stripes at the same time, at least in single-redundancy arrays.[13] Thus, it is possible, in cases where only one storage block is in the "association," only one stripe would be referenced by that "association." However, because the claim requires an "association" capable of mapping data blocks to storage blocks "across the plurality of stripes," one of ordinary skill in the art would understand that the term "one or more" covers a degenerate (abnormal) case only, because, in practical terms, using more than one stripe, and therefore more than one storage block, is an important feature of the data structure being defined as "association". Ganger Decl. ¶ 94.

Finally, one of ordinary skill in the art would understand that the phrase "associating the data blocks" refers to file system blocks as opposed to "storage blocks". One of ordinary skill would understand that the two types of blocks need not be the same size, and that the claim therefore does not require a one-to-one mapping between the data blocks and the one or more storage blocks. That this one-to-one mapping is not required is confirmed by the plain language of other claims. For instance, claim 1, reads in relevant part "associating ***each** data block* with *a respective* ***one*** of the storage blocks…" Ho Decl., Exh C. ('715 patent) at 20:7-8. That the inventors claimed this type of one-to-one mapping in other claims is evidence that they knew how to do so and that claims lacking this language of limitation should therefore not be read to include it.

[13] Sun notes in its brief that, "In his recent deposition, one of the named inventors on the '715 patent, Douglas Doucette, acknowledged a given storage block is not part of two different stripes at the same time." Sun's characterization of Mr. Doucette's testimony is misleading in that it relates to a comment he made relative to implementations of NetApp's products prior to the release of the first product that NetApp contends embodies the subject matter claimed in the '715 patent. Ho Decl., Exh. I, (NetApp's P.R. 3-1 Disclosure for the '715 Patent) at 2; Williamson Decl., Exh. K (Doucette Depo. at 133).

b. If the Court Deems A Construction Necessary, It Should Adopt NetApp's Alternative Construction

While NetApp does not believe that this term needs any clarification because its meaning will be clear to the jury as written, NetApp has nevertheless provided a possible construction for each of these disputed terms to assist the jury if the Court determines such assistance is needed. NetApp's proposed constructions assist the jury because they accurately convey (1) that there is a data structure called an "association" that associates data blocks (from the file system) with storage blocks (the file system's representation of space in the RAID), and (2) that, in the normal case, blocks from multiple stripes are used in each "association." As set forth below, the specification and file history confirm this understanding of the disputed claim terms.

(1) The Specification Confirms That An "Association" Is A Data Structure

As an initial example, the specification calls for *both* data blocks *and* an "association" to be transmitted, thus showing that the association is an actual data structure and not an abstraction. *See, e.g.*, Ho Decl., Exh C. ('715 patent) at 13:14-23 ("The *data blocks and the association* are transmitted to, and processed by, the disk array manager 13 so that each data block is stored **at** its associated storage block in the group 120."). Requiring that the "association" – and not just the raw data blocks – be transmitted to the storage devices, this statement confirms that the "association" is an actual thing, to wit, a data structure. Indeed, the "association" *per se* could not be "transmitted" unless it was a *bona fide* data structure. In order for the storage system that receives the transmission to understand the association, the data is organized as a data structure. *See* Ganger Decl. ¶ 96-97. Similar statements buttressing this understanding appear throughout the specification:

- The specification explains that the "association" can take on "alternative embodiments," demonstrating that the patent does not use the term "association" to refer to an abstraction, but a genuine data structure. *See* Ho Decl., Exh C ('715 patent) at 13:34-43 ("In *alternative embodiments of an association* 15A, not all free storage blocks in the group of storage blocks are associated with buffered data blocks.").
- The specification uses the term "association" as a concrete noun and equates it to a "RAID map," which is well understood in the art to be a data structure. *See* Ho Decl., Exh C ('715 patent) at 16:20-26. ("The association of a range or

ranges of VBN's to objects at each level is sometimes referred to as a RAID map.").

- FIG. 2 of the patent depicts a disk array sending a two dimensional grid (i.e., the "association") to the disk array manager. The rectangular grid being sent from the file system to the array is not undifferentiated, but has multiple discrete cells, thus depicting a genuine data structure *containing* data in the separate cells of the grid, rather than the mere passing of the data itself. *See* Ho Decl., Exh. C ('715 patent) at Fig. 2; *see also id.* at Fig. 8 (depicting the "association" in even greater detail).

The above makes clear that the "association" is more than mere verbiage to describe a buffer of data blocks being passed from one part of the depicted system to another, but an entity that exists in its own right as a distinguishable limitation in the claims. Ho Decl., Exh C. ('715 patent).

(2) **The Specification Confirms That Blocks From Multiple Stripes Are Typically Used In Each "Association"**

The specification specifically distinguishes the prior art on the basis of the prior art not generally carrying out writes to multiple stripes. Indeed, the specification explains that "[p]rior art systems, in contrast, typically send single stripe write transactions to a RAID…" Ho Decl., Exh C ('715 patent) at 9:28-29. The specification explains, for example, that "The method includes writing data to a group of storage blocks that include predetermined storage blocks across a plurality of stripes…" *Id.* at 3:14-16. Further evidence that "association" data structures typically include blocks from multiple stripes is located throughout the '715 patent, at, *e.g.*, 3:29-31; 3:66-4:2; 5:54-58; and 3:7-8. *See also*, FIG. 8, providing a detailed view of an association including data to be written to multiple stripes in a single write request.

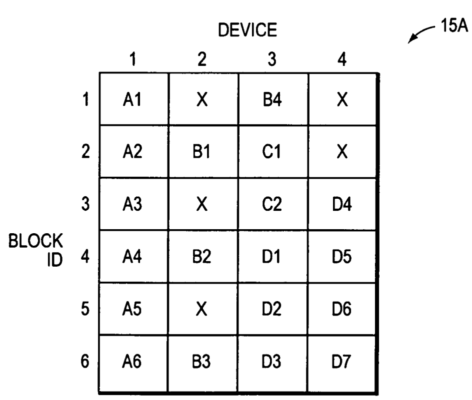

FIG. 8

Thus, the ’715 invention, unlike the prior art, is directed to optimized *multiple* stripe writes. But, in the ’715 invention, a multiple stripe write strictly requires an “association” that includes blocks from multiple stripes. The claims should be thus be so understood.

c. The Disputed Terms Are Not Indefinite

Sun’s argument on this issue is nothing more than a specious grammar debate, and indefiniteness requires much more than that:

> We have not insisted that claims be plain on their face in order to avoid condemnation for indefiniteness; rather, what we have asked is that the claims be amenable to construction, however difficult that task may be. If a claim is insolubly ambiguous, and no narrowing construction can properly be adopted, we have held the claim indefinite. If the meaning of the claim is discernible, even though the task may be formidable and the conclusion may be one over which reasonable persons will disagree, we have held the claim sufficiently clear to avoid invalidity on indefiniteness grounds.

Exxon Research & Eng'g Co. v. United States, 265 F.3d 1371, 1374 (Fed. Cir. 2001). Sun’s argument does not even come close to meeting this stringent test. Briefly, Sun’s argument is that because the terms allegedly require a single storage block to be written on more than one stripe at the same time, they are indefinite. Specifically, Sun argues that under the ’715 patent, this would be “impossible”. *See* Sun’s Opening Cl. Constr. Br.at 33. One of ordinary skill in the art would understand that a storage block, in the context of the ’715 patent means the part of a stripe occupying an individual disk in an array. Ganger Decl. ¶ 93-94; Ho Decl., Exh. C (’715 patent) at 1:37-39, (“Blocks of data are written to a RAID system in the form of stripes, where a stripe includes one storage block on each disk drive in an array of disk drives in the system.”).[14] Furthermore, one of ordinary skill in the art would understand that one storage block would not occupy multiple stripes at the same time. Rather, one of ordinary skill in the art would understand one or both of two consistent things: First, that the storage blocks exist across a plurality of stripes (as clearly described in the claims themselves), and second, that to the extent a sufficient amount of data blocks are to be written, storage blocks will be selected for that purpose across more than one stripe. Naturally, while a major purpose of the ’715 patent is to optimize RAID I/O by a file system – including by writing to more than one stripe in the same request –

[14] While there is no requirement that a stripe include one block per disk in the RAID, this is a standard implementation.

the inventors wisely chose not to exclude from the scope of their claims a write request that is too small to occupy blocks on multiple stripes, which is why the claims refer to "one or more blocks across the plurality of stripes." This language claims what is intended as a significant advance in I/O optimization, while allowing that the system must still function in the rare instance where there is not enough data to warrant this type of optimization.

Sun's argument is analogous to an argument that the sentence, "We will eat in one or more McDonald's across the United States," would not enable a speaker of ordinary English to know the plan. According to Sun, the phrase, "one or more McDonald's across the United States" would be indefinite, specifically requiring that at least in some cases a single McDonald's must exist in more than one state at the same time. In fact, the average McDonald's diner of our analogy, and certainly someone of skill in the art of the '715 patent, would understand the following: "We are *planning* to eat at more than one McDonalds, and *if* we do, they will *probably* be in more than one state—but it's *possible* we will only eat at one McDonalds."

Likewise, the disputed claim language is not difficult to understand. It establishes that there are storage blocks across a plurality of stripes. It establishes that the plan is to associate data blocks with one or more of the stripes. To the extent possible, it will do so across the plurality of stripes.

d. Sun's Proposed Construction Is Wrong And Is Derived From A Mischaracterization Of The File History

Sun's construction impermissibly reads limitations from unasserted claims and misuses the file history to attempt to add limitations into the asserted claims. To support its construction, Sun mischaracterizes the file history. Since at least *Phillips*, the Federal Circuit has warned against the dangers of too hastily presuming that every statement in the prosecution history is automatically the kind of "clear and unmistakable surrender of claim scope," necessary in order to apply a narrowing construction:

> Because of the potential for such ambiguities, we have recognized that "because the prosecution history represents an ongoing negotiation between the PTO and the applicant ... it often lacks the clarity of the specification and thus is less useful for claim construction purposes." *Elbex Video, Ltd. v. Sensormatic Electronics Corp.* 508 F.3d 1366, 1372 (Fed. Cir. 2007), citing *Phillips v. AWH Corp.,* 415 F. 3d 1303, 1317 (Fed. Cir. 2005).

Sun argues in its opening brief that, "NetApp repeatedly argued to the Examiner that the claims of the '715 patent were distinguishable over the prior art because the claims require associating each data block with a respective one of the storage blocks. This construction [sic]… enabled NetApp to obtain the '715 patent…" Sun's Opening Cl. Constr. Br. at 35. Sun further argues that, "This construction [sic]… enabled NetApp to obtain the '715 patent. *Id.* Thus Sun urges that the so-called "one-to-one" mapping of data blocks to storage blocks, which appears in its proposed construction, was a necessary point of novelty. In fact, the opposite is true. As NetApp's statements to the examiner reveal, and as explained below, NetApp viewed writes to a "plurality of stripes" and the "association" data structure as innovations. In fact, NetApp did not even include the so-called "one-to-one" mapping terminology in many of its pending claims—including those asserted here. Miscasting the prosecution history, Sun incorrectly insists that NetApp narrowly characterized a claim limitation, and thereby created a clear and unambiguous disavowal of the broader claim scope to overcome prior art. What actually happened, as the file history reveals, is that the examiner *erroneously* found that a limitation recited in certain claims – which are not asserted here– was present in a certain prior art reference. NetApp repeatedly argued that the prior art did *not* contain the limitation, but never stated or suggested that the limitation was a requirement of every single claim in the pending application.

In particular, the examiner rejected certain claims on the grounds that the prior art DeKoning reference "teaches… associating each data block with a respective one of the storage blocks, for transmitting the association to a storage device manager for processing of the single write transaction." Ganger Decl., Exh. U (Office Action, May 20, 2004) at 3. In response, the applicants distinguished DeKoning from "representative" application claim 17 as follows: "the DeKoning patent is legally precluded from anticipating the claimed invention under 35 U.S.C. § 102 because of the absence … of Applicant's 'associating each data block with a respective one of the storage blocks, for transmitting the association to a storage device manager for processing of the single write transaction.'" Ganger Decl., Exh. V (Amendment Aug. 20, 2004) at 17; *see also*, *id.* at 15 (Where NetApp urged that, "all art cited during prosecution… is completely silent

regarding '***associating each data block with a respective one of the storage blocks***' as claimed." (emphasis in original)). More broadly, applicants argued that "DeKoning does not address associating data blocks with storage blocks, but instead merely discusses the use of buffering smaller write requests into a larger write request, such as a RAID stripe write."[15] *Id.* at 18. Thus, the point of distinction was not the supposed one-to-one relationship (which was relevant to original claim 17), but instead the broader "association" concept as well as the concept of selecting blocks, where possible, from across the plurality of stripes. NetApp actually argued, "Applicant goes one step further by mapping each data block of the single write request with a storage block across a ***plurality of stripes***...." *See* Ho Decl., Exh. L (Preliminary Amendment, Feb. 22, 2005) at 20 (emphasis in original). Sun presents the court with its own version of this quotation as follows: "**Applicant goes one step further by mapping each data block of the single write request with a storage block** *across the plurality of stripes*..." *See* Sun's Opening Cl. Constr. Br. at 36. Thus, in Sun's description of the file history, Sun radically alters the emphasis in order to make it appear as though the so-called "one-to-one" mapping were the point of novelty.[16]

At base, Sun is arguing that every *potential* distinction over the prior art – whether genuine or not – must be found in every single allowed claim, regardless of whether there is any linguistic support in a specific claim. This argument is without merit. When read in context, the file history makes clear that applicants were relying on the "association" and the "plurality of

---

[15] Notably the examiner eventually agreed with NetApp, admitting that, "The prior art does not suggest mapping each data block with a respective storage block across the plurality of stripes and transmitting that mapping to the storage device manager for processing a single write request." Ho Decl., Exh. J (Office Action, May 17, 2005) at 2-3, thereby admitting a broad combination of things *not* found in the prior art.

[16] It is worth noting that the various mapping terms found in the patent including, for example, "each… to a respective one," "associating… with one or more," and "each… with a single," (*e.g.* Ho Decl., Exh. C) at 21:36, 21:47-49, 21:55-56) all grammatically permit multiple possible associations: one to one; one to many; many to one; many to many. In addition, it is noteworthy that the examiner's explanation of reasons for allowance did not isolate a single novel feature but listed several features together, and thus does not give rise to any inference as to a specific point of novelty: "…the prior art does not further teach buffering write requests, associating each data block to be stored with a respective one of the storage blocks across the plurality of stripes for a single write operation, and transmitting this association to the storage device manager…" Ho Decl., Exh. K (Notice of Allowability, Oct. 28, 2005) at 2.

stripes" as absent from the cited reference. By doing so, the applicants did not narrow the claims to the one-to-one relationship now sought by Sun. Certainly, they did not make the clear and unambiguous disclaimer that the law requires for an estoppel by argument to attach. *See, e.g.*, *Omega Eng'g Inc. v. Raytek Corp*., 334 F.3d 1314, 1326 (Fed. Cir. 2003). "For prosecution disclaimer to attach, our precedent requires that the alleged disavowing actions or statements made during prosecution be both clear and unmistakable."); *Elkay Mfg. Co. v. Ebco Mfg. Co.*, 192 F.3d 973, 979 (Fed. Cir. 1999) ("To determine whether an argument acts as estoppel, the court should assess 'the totality of the prosecution history, not the individual segments of the presentation made to the Patent and Trademark Office by the applicant.'"). The file history not only shows that no such "clear and unmistakable" action or statement was taken; and furthermore that Sun's interpretation of the file history attempts to isolate the "individual segments" rather than to shed light on the totality of the prosecution history. Sun's proposed construction alters the plain meaning of the claims at issue to avoid clarifying the important limitation that there actually be a *bona fide* data structure capable of mapping file system blocks to storage blocks, and to add another limitation of one-to-one mapping which is found nowhere in the asserted claims.

III.

CONCLUSION

For the aforementioned reasons, the Court should adopt NetApp's proffered constructions of the disputed claim terms.

Dated: July 21, 2008

WEIL, GOTSHAL & MANGES LLP

By: */s/ Edward R. Reines*
Edward R. Reines
Attorneys for
Plaintiff-Counterclaim Defendant
NETAPP, INC.